Bayes Risk minimization using metric loss functions

Schlüter, Ralf ; Scharrenbach, Thomas ; Steinbiss, Volker ; Ney, Hermann

Abstract: In this work, fundamental properties of Bayes decision rule using general loss functions are derived analytically and are verified experimentally for automatic speech recognition. It is shown that, for maximum posterior probabilities larger than 1/2, Bayes decision rule with a metric loss function always decides on the posterior maximizing class independent of the specific choice of (metric) loss function. Also for maximum posterior probabilities less than 1/2, a condition is derived under which the Bayes risk using a general metric loss function is still minimized by the posterior maximizing class. For a speech recognition task with low initial word error rate, it is shown that nearly 2/3 of the test utterances fulfil these conditions and need not be considered for Bayes risk minimization with Levenshtein loss, which reduces the computational complexity of Bayes risk minimization. In addition, bounds for the difference between the Bayes risk for the posterior maximizing class and minimum Bayes risk are derived, which can serve as cost estimates for Bayes risk minimization approaches.

Bayes Risk Minimization using Metric Loss Functions

R. Schlüter, T. Scharrenbach, V. Steinbiss, H. Ney

Lehrstuhl für Informatik 6 - Computer Science Dept.
RWTH Aachen University, Aachen, Germany
{schlueter,steinbiss,ney}@cs.rwth-aachen.de

Abstract

In this work, fundamental properties of *Bayes* decision rule using general loss functions are derived analytically and are verified experimentally for automatic speech recognition. It is shown that, for maximum posterior probabilities larger than $1/2$, *Bayes* decision rule with a metric loss function always decides on the posterior maximizing class independent of the specific choice of (metric) loss function. Also for maximum posterior probabilities less than $1/2$, a condition is derived under which the *Bayes* risk using a general metric loss function is still minimized by the posterior maximizing class. For a speech recognition task with low initial word error rate, it is shown that nearly 2/3 of the test utterances fulfil these conditions and need not be considered for *Bayes* risk minimization with *Levenshtein* loss, which reduces the computational complexity of *Bayes* risk minimization. In addition, bounds for the difference between the *Bayes* risk for the posterior maximizing class and minimum *Bayes* risk are derived, which can serve as cost estimates for *Bayes* risk minimization approaches.

1. Introduction

In speech recognition, the standard evaluation measure is word error rate (WER). On the other hand, the standard decision rule for speech recognition (maximization of the sentence posterior probability) is realized by using a sentence error based (or 0-1) cost function for *Bayes* decision rule. Due to the complexity of the *Levenshtein* alignment needed to compute the number of word errors, it was prohibitive to use the number of word errors as cost function for *Bayes* decision rule for a long time. Nevertheless, with the constant increase in computing power and with algorithmic improvements, word error minimizing decision rules became more realistic. Consequently, a number of approaches were presented in literature which investigate efficient approximate realizations of word error minimizing *Bayes* decision rules. In these approaches, approximations were done at different levels: search space, summation space for expected loss calculation, and the loss function itself. In [5], the search space as well as the expected loss calculation were reduced to N-best lists. In [1, 2], the search space is represented by word graphs and the expected loss calculations are performed on the tree of partial hypotheses which define the stack of an A* search with a specific choice of cost estimates. In [4], the search space/summation space for expected risk calculation is approximated by consensus lattices, for which the expected loss calculation as well as the search become much more efficient. Finally in [6], the cost function itself is modified, i.e. the word error cost is replaced by a frame-wise word error based on forced alignments. In [6], it can be observed that the relative improvements obtained with a word error minimizing decision rule increase with the baseline error rates.

Of all these approaches, the word graph based A* search for word error minimization presented in [1, 2] is closest to the correct *Bayes* decision rule. In fact, the method presented in [1, 2] would be exact if it was not for the necessity of pruning, which, in this case, does not only reduce the search space as usual, but also has an effect on the decision within the remaining search space, as will be shown in this work (cf. Section 3.5).

In literature, word error minimizing *Bayes* decision rules are often called "Minimum *Bayes* Risk". This is somewhat misleading, since the standard approach also minimizes the *Bayes* risk - but by using a sentence error or 0-1 loss function. The important difference lies in the cost function used, which usually counts word errors or sentence errors.

In this work, *Bayes* decision rule is analyzed on a more fundamental level. We present a number of properties of the *Bayes* decision rule, mainly concerning the relation between using a 0-1 loss function (e.g. sentence errors) and a general loss function, such as phoneme/character/word errors in speech recognition and machine translation, or position independent word error in machine translation, to name but a few. We present analytic results, simulations, as well as speech recognition results for a small and large vocabulary.

The remainder of this work is organized as follows. After a general introduction to *Bayes* decision rule in Sec. 2, we derive bounds for the difference in *Bayes* risk between a 0-1 and a general loss function in Sec. 3. In Sec. 3 we also show that, under certain conditions, the decisions with 0-1 and with general loss function are identical, for one of the conditions even independent of the explicit choice of general (metric) loss function. It is shown that, in some cases, the class posterior distribution dominates the decision, and in other cases, the structure of losses dominates the decision. In Sec. 4, we provide experimental evidence of the analytic results derived for the case of automatic speech recognition.

2. Bayes Decision Theory

Consider the class posterior distribution $p(c|x)$ for classes c given an observation x. Since the derivations given in this work do not depend directly on the specific choice of class and observation, for simplicity all considerations and derivations are given using abstract classes and observations unless otherwise specified. All derivations are valid for complex class definitions like word sequences in speech recognition and machine translation.

In its general form, i.e. using a general loss function $\mathcal{L}(c, c')$, *Bayes* decision rule results in the class minimizing the expected loss:

$$\begin{aligned} r_{\mathcal{L}}(x) &= \operatorname*{argmin}_{c} \sum_{c'} p(c'|x)\mathcal{L}(c', c) \\ &= \operatorname*{argmin}_{c} \mathcal{R}_{\mathcal{L}}(c), \end{aligned}$$

with the expected loss, or *Bayes* risk $\mathcal{R}_{\mathcal{L}}(c)$ for class c:

$$\mathcal{R}_{\mathcal{L}}(c) := \sum_{c'} p(c'|x)\mathcal{L}(c', c).$$

In particular using the 0-1 loss function, *Bayes* decision rule can be reduced to finding the class which maximizes the class posterior probability:

$$r_{0-1}(x) = \operatorname*{argmax}_c p(c|x).$$

Due to complexity reasons, in speech recognition and machine translation the *Bayes* decision rule based on the 0-1 loss function is usually applied, i.e. the sentence error rate is usually minimized. Therefore in the following, a number of general properties of *Bayes* decision rule using a general loss function in contrast to using a 0-1 loss function are derived.

For simplicity, in the following, we will drop the observation- or x-dependence of the posterior probability and use c_{max} for the class which maximizes the class posterior probability, and $c_{\mathcal{L}}$ for the class which minimizes the *Bayes* risk for loss function $\mathcal{L}$.

3. Analysis of Bayes Decision Rule with General Loss Functions

In the following, general properties of *Bayes* decision rule will be presented for the case of general loss functions which fulfil the properties of a metric. A metric loss function is positive, symmetric, and fulfils the triangle inequality. A metric loss function is zero if and only if both arguments are equal.

3.1. Loss-Independence of the *Bayes* Decision Rule for Large Posterior Probabilities

Assume a maximum posterior probability $p(c_{\text{max}}) \geq \frac{1}{2}$ and a metric loss $\mathcal{L}(c, c')$. Then the posterior maximizing class c_{max} also minimizes the *Bayes* risk.

Proof: Consider the difference between the *Bayes* risk for class c_{max} and the *Bayes* risk for any class c':

$$\begin{aligned}
&\mathcal{R}_{\mathcal{L}}(c_{\text{max}}) - \mathcal{R}_{\mathcal{L}}(c') = \sum_c p(c)\mathcal{L}(c, c_{\text{max}}) - \sum_c p(c)\mathcal{L}(c, c') \\
&= -\underbrace{p(c_{\text{max}})}_{\geq \frac{1}{2} \geq \sum_{c \neq c_{\text{max}}} p(c)} \mathcal{L}(c_{\text{max}}, c') + \sum_{c \neq c_{\text{max}}} p(c) \left[\mathcal{L}(c, c_{\text{max}}) - \mathcal{L}(c, c')\right] \\
&\leq -\sum_{c \neq c_{\text{max}}} p(c) \underbrace{\left[\mathcal{L}(c, c') + \mathcal{L}(c', c_{\text{max}}) - \mathcal{L}(c, c_{\text{max}})\right]}_{\geq 0 \quad \text{(triangle inequality)}} \\
&\leq 0.
\end{aligned} \tag{1}$$

It can be shown that the *Levenshtein* distance function fulfils the properties required for a metric. Therefore the above proof, among others, is valid for e.g. phoneme, character, and word error loss functions. The same applies to the position independent word error rate, provided that classes are word sets instead of sequences and that the posterior probabilities for permutations of a given word sequence are summed up to produce the posterior probability of a word set.

Using the above derivation for large posterior probabilities, a broad estimate can be calculated, which shows where word error minimization can be expected to result in different decisions than sentence error minimization or posterior probability maximization respectively. Assume a task has an expected word error rate of r. In addition assume the words in a sentence to be statistically independent. Then the average posterior probability p for a sentence of length M would be $p = (1-r)^M$. From Eq. (1), we know that only for maximum posterior probabilities $p(c_{\text{max}}) < 1/2$ we can expect a difference between sentence and word error minimization. Therefore, if word error minimization is to make any difference, the expected word error rate r needs to fulfil the following approximate inequality $r \gtrsim 1 - (1/2)^{\frac{1}{M}}$. For example, if a sentence is 18 words long, the word error rate needs to be larger than 4% so that word error minimization makes some difference.

3.2. Dominance of Maximum Posterior Probability

Now we assume the maximum posterior probability is less than $1/2$ and the loss function $\mathcal{L}(c, c')$ is a metric. Then the class maximizing the posterior probability also minimizes the *Bayes* risk if a set $\mathcal{C}$ of classes can be found for which the following requirements are met:

$$c_{\text{max}} \notin \mathcal{C}$$

$$\sum_{c \in \mathcal{C}} p(c) \geq 1 - 2p(c_{\text{max}}) + \max_{c \in \mathcal{C}} p(c) \tag{2}$$

$$\mathcal{L}(c, c_{\text{max}}) \leq \mathcal{L}(c, c') \quad \forall \quad c, c' \in \mathcal{C}. \tag{3}$$

Proof: Consider again the difference between the *Bayes* risk for class c_{max} and the *Bayes* risk for any class c':

$$\begin{aligned}
&\mathcal{R}_{\mathcal{L}}(c_{\text{max}}) - \mathcal{R}_{\mathcal{L}}(c') = \sum_c p(c)\mathcal{L}(c, c_{\text{max}}) - \sum_c p(c)\mathcal{L}(c, c') \\
&= \left[p(c') - p(c_{\text{max}})\right] \mathcal{L}(c_{\text{max}}, c') \\
&\quad + \sum_{c \neq c_{\text{max}}, c'} p(c) \left[\mathcal{L}(c_{\text{max}}, c) - \mathcal{L}(c', c)\right] \\
&\overset{(2)}{\leq} \sum_{c \notin \mathcal{C} \cup \{c_{\text{max}}\}} \mathcal{L}(c_{\text{max}}, c') \\
&\quad + \sum_{c \neq c_{\text{max}}, c'} p(c) \left[\mathcal{L}(c_{\text{max}}, c) - \mathcal{L}(c', c)\right] \\
&= -\sum_{c \notin \mathcal{C} \cup \{c_{\text{max}}\}} p(c) \underbrace{\left[\mathcal{L}(c, c') + \mathcal{L}(c', c_{\text{max}}) - \mathcal{L}(c, c_{\text{max}})\right]}_{\geq 0 \quad \text{(triangle inequality)}} \\
&\quad + \sum_{c \in \mathcal{C} \setminus \{c'\}} p(c) \left[\mathcal{L}(c_{\text{max}}, c) - \mathcal{L}(c', c)\right] \\
&\leq \sum_{c \in \mathcal{C} \setminus \{c'\}} p(c) \left[\mathcal{L}(c_{\text{max}}, c) - \mathcal{L}(c', c)\right] \overset{(3)}{\leq} 0.
\end{aligned} \tag{4}$$

As a special case, condition (3) is fulfilled if the following inequality is fulfilled:

$$\mathcal{L}(c, c_{\text{max}}) \leq 1 \quad \forall \quad c \in \mathcal{C}. \tag{5}$$

Note that Ineq. (5) can be checked much more efficiently than Ineq. (3). Therefore, the efficiency of *Bayes* risk minimization using a general, non- 0-1 loss function can be improved by using Ineq. (1), and Ineq. (4) together with Ineq. (5) to shortlist those test samples, for which a difference to using a 0-1 loss function can be expected. All remaining samples can be classified using the more efficient 0-1 loss function.

3.3. Upper Bounds of Risk Difference

In the case of a maximum posterior probability $p(c_{\text{max}}) < \frac{1}{2}$ and a metric loss $\mathcal{L}(c, c')$, the following upper bound can be derived

for the difference between the *Bayes* risk for class $c_{\max}$ and the *Bayes* risk for any class c':

$$\begin{aligned}
&\mathcal{R}_{\mathcal{L}}(c_{\max}) - \mathcal{R}_{\mathcal{L}}(c') \\
&= \sum_{c} p(c) \left[\mathcal{L}(c, c_{\max}) - \mathcal{L}(c, c')\right] \\
&= p(c')\mathcal{L}(c', c_{\max}) - p(c_{\max})\mathcal{L}(c_{\max}, c') \\
&\quad + \sum_{c \neq c_{\max}, c'} p(c) \underbrace{\left[\mathcal{L}(c, c_{\max}) - \mathcal{L}(c, c')\right]}_{\leq \mathcal{L}(c_{\max}, c') \quad \text{(triangle ineq.)}} \\
&\leq \left[1 - 2p(c_{\max})\right] \mathcal{L}(c_{\max}, c').
\end{aligned} \tag{6}$$

Replacing c' by the class $c_{\mathcal{L}}$ which minimizes the *Bayes* risk, then Ineq. (6) leads to the following upper bound:

$$\mathcal{R}_{\mathcal{L}}(c_{\max}) - \mathcal{R}_{\mathcal{L}}(c_{\mathcal{L}}) \leq \left[1 - 2p(c_{\max})\right] \mathcal{L}(c_{\max}, c_{\mathcal{L}}). \tag{7}$$

For the following specific choice, Ineq. (7) can be shown to be tight:

$$\mathcal{L}(c, c_{\mathcal{L}}) = 1 \quad \forall \quad c \in \mathcal{C}, \tag{8}$$

$$\mathcal{L}(c, c_{\max}) = \mathcal{L}(c_{\max}, c_{\mathcal{L}}) + 1 \quad \forall \quad c \in \mathcal{C}. \tag{9}$$

$$\mathcal{L}(c, c') = \mathcal{L}(c_{\max}, c_{\mathcal{L}}) + 1 \quad \forall \quad c, c' \in \mathcal{C}, \tag{10}$$

Note that the special choice made in Eqs. (8-10) in general are not realizable for all combinations of vocabulary size, string length, loss function, and maximum posterior probability $p(c_{\max})$. In the case of $\frac{1}{3} \leq p(c_{\max}) \leq \frac{1}{2}$, a single element $c \in \mathcal{C}$ is sufficient to show the tightness of the derived upper bound. The upper bound can be reached with the choice $p(c_{\mathcal{L}}) = p(c_{\max})$, $p(c) = 1 - 2p(c_{\max})$ and $\mathcal{L}(c, c_{\max}) - \mathcal{L}(c, c_{\mathcal{L}}) = \mathcal{L}(c_{\max}, c_{\mathcal{L}})$. There are also examples for the upper bound not being tight. For strings of length N and a loss function only allowing substitutions, the loss cannot exceed N. For $\mathcal{L}(c_{\max}, c_{\mathcal{L}}) = N$, condition 9 cannot be fulfilled since it would require $\mathcal{L}(c, c_{\max}) = N + 1$, which cannot occur in the case of such a loss function.

Another upper bound can be found using the following definition of the set of classes $\mathcal{C}_\epsilon$:

$$\mathcal{C}_\epsilon := \{c | \mathcal{L}(c, c_{\max}) \leq \epsilon\} \tag{11}$$

with

$$\epsilon := \min\left\{\lambda \Big| \sum_{c' : \mathcal{L}(c_{\max}, c') \leq \lambda} p(c') \geq \frac{1}{2}\right\} \tag{12}$$

For the case $c' \notin \mathcal{C}_\epsilon$ the following inequality can be derived:

$$\begin{aligned}
&\mathcal{R}_{\mathcal{L}}(c_{\max}) - \mathcal{R}_{\mathcal{L}}(c') = \sum_{c} p(c) \left[\mathcal{L}(c, c_{\max}) - \mathcal{L}(c, c')\right] \\
&= -p(c_{\max})\mathcal{L}(c_{\max}, c') \\
&\quad + \sum_{c \in \mathcal{C}_\epsilon \setminus c_{\max}} p(c) \underbrace{\left[\mathcal{L}(c, c_{\max}) - \mathcal{L}(c, c')\right]}_{\leq 2\epsilon - \mathcal{L}(c_{\max}, c') \quad \text{(triangle ineq.) and Eq. (11)}} \\
&\quad + \sum_{c \notin \mathcal{C}_\epsilon} p(c) \underbrace{\left[\mathcal{L}(c, c_{\max}) - \mathcal{L}(c, c')\right]}_{\leq \mathcal{L}(c_{\max}, c') \quad \text{(triangle ineq.)}} \\
&\geq \underbrace{\left[1 - 2\sum_{c \in \mathcal{C}_\epsilon} p(c)\right]}_{\geq 0} \underbrace{\mathcal{L}(c_{\max}, c')}_{\geq \epsilon} + 2\epsilon \Big[\sum_{c \in \mathcal{C}_\epsilon} p(c) - p(c_{\max})\Big] \\
&\geq \left[1 - 2p(c_{\max})\right] \epsilon \quad \forall \quad c' \notin \mathcal{C}_\epsilon.
\end{aligned}$$

For the case of $c' \in \mathcal{C}_\epsilon$ we can apply Ineq. (6) to obtain the same inequality. Therefore we obtain:

$$\mathcal{R}_{\mathcal{L}}(c_{\max}) - \mathcal{R}_{\mathcal{L}}(c') \leq \left[1 - 2p(c_{\max})\right] \epsilon \quad \forall \quad c'. \tag{13}$$

Note that $\mathcal{L}(c_{\max}, c_{\mathcal{L}})$ is needed for Ineq. (6), for which word error minimization would have to be performed. In contrast to this, ϵ can be found efficiently, i.e. with complexity linear in the number of classes. Ineq. (13) can be used to delimit overestimates as they are used in A*-based *Bayes* risk minimization approaches as presented in [1, 2].

3.4. Minimum Risk with Zero Posterior Probability

Consider the example shown in Fig. 1, where four character strings are shown at the nodes of the graph. The *Levenshtein* loss is given at the arcs of the graph, and the posterior probabilities of the sequences are shown at the nodes. The *Bayes* risk for the four strings then gives:

$$\begin{aligned}
\mathcal{R}_{\mathcal{L}}(\text{ded}) &= 1 & \mathcal{R}_{\mathcal{L}}(\text{aded}) &= 2(1 - p_2) \\
\mathcal{R}_{\mathcal{L}}(\text{dgd}) &= 2(1 - p_1) & \mathcal{R}_{\mathcal{L}}(\text{dedb}) &= 2(1 - p_3).
\end{aligned}$$

Provided all posterior probabilities are less than $1/2$, i.e. $p_i < 1/2 \ \forall \ i = 1, 2, 3$, the loss for string "ded" will be minimal even though its posterior probability is zero[1]. From the example it becomes clear that "ded" wins since it is the most *consistent* hypotheses.

3.5. Pruning and General Loss Functions

In [1, 2] an approach to word error minimization is presented, which uses A* search and cost estimates using partial hypotheses to find the sentence hypotheses minimizing the expected word error rate. In this approach, the search space and the summation space for the expected loss calculation are the same. Without pruning this approach still is exact. In the following we will show that pruning not only reduces the search space but also alters the decision for the remaining search space. Consider the example from Section 3.4 with the following posterior probabilities: $p_1 = \frac{2}{9}$, $p_2 = \frac{3}{9}$, and $p_3 = \frac{4}{9}$. If we apply the algorithm presented in [1, 2] to the example, then without pruning, we obtain the correct class "ded" which minimizes the *Bayes* risk using the *Levenshtein* loss. But if we prune the worst hypothesis (with respect to *Bayes* risk) "dgd" at some stage of the word error minimizing search, then the result would be "dedb" instead, which would be correct for the remaining subspace as indicated in [1, 2], but it would be incorrect with respect to the complete space.

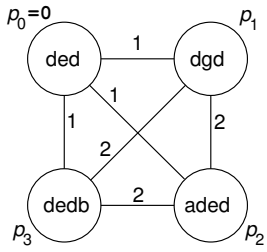

Figure 1: *Example for the case of a string with zero posterior probability ("*ded*") resulting in minimum* Bayes *risk. The arcs of the graph show the-based Levenshtein distance between all strings shown.*

[1] In practice a zero posterior probability for "ded" could mean that it has been pruned before applying word error minimization.

3.6. Simulations

Bayes risk minimization using stochastic simulations of fully dependent posterior distributions for the cases of word sequences with *Levenshtein* based word error loss function and for word sets with position independent word error loss were performed. Due to the exponential complexity of the simulations, only low sequence lengths/set cardinalities as well as small vocabulary sizes were considered. All results obtained were consistent with the analytic results presented.

4. Experiments

To verify the analytic results derived in Sec. 3, we performed speech recognition experiments using the WSJ0 corpus with a vocabulary of 5k words. The baseline recognition system [3] uses 1500 generalized triphone states plus one silence state, *Gaussian* mixture distributions with a total of about 147k densities with a single pooled variance vector. 33-dimensional observation vectors are obtained by Linear Discriminant Analysis (LDA) based on 5 consecutive vectors consisting of 12 MFCCs, their first derivatives and the second derivative of the energy, which are extracted at a 10ms frame shift. The system was trained on the WSJ0 training corpus (15h speech) and a trigram language model was used. The baseline word error rate (WER) is 3.97% on the ARPA WSJ0 Nov. '92 corpus using the standard decision rule maximizing the sentence posterior probability.

The experiments for word error minimization were performed using N-best lists [5] with $N = 10,000$ to ensure proper normalization of the posterior probabilities. The search for the minimum *Bayes* risk using the *Levenshtein* loss function was always started of by calculating the risk for the posterior maximizing word sequence first, which served as an initial risk pruning threshold. Note that pruning here is only performed on the search space to stop the summation for a hypotheses once the risk exceeds the risk for an already existing hypothesis. In Table 1 the recognition results using *Bayes* decision rule with 0-1 loss (sentence error minimization) and with *Levenshtein* loss (word error minimization) are summarized. In 54% of the utterances the maximum posterior probability is greater or equal to $1/2$, i.e. the decision is the same for 0-1 and *Levenshtein* loss function, as shown in Sec. 3.1. This is also the case in another 8% of the utterances where the Ineqs. (2) and (5) hold. Hence, for nearly 2/3 of the utterances, *Bayes* risk minimization is proven to result in the posterior maximizing class! Therefore, word error minimization here would have to be performed for only about 1/3 of the utterances, which reduces the computational complexity. Here, it can also be observed that word error minimization only gives a marginal improvement in word error rate from 3.97% for sentence error minimization down to 3.88% for word error minimization.

Table 1: *Analysis of word error minimization on the ARPA WSJ0 Nov. '92 corpus. Results are presented for sub-corpora based on the following conditions: a) $p(c_{max}) \geq 1/2$ (cf. Sec. 3.1), b) Ineqs. (2) and (5) are fulfilled (cf. Sec. 3.2), c) $c_{max} = c_{\mathcal{L}}$ but neither a) nor b) hold, d) $c_{max} \neq c_{\mathcal{L}}$.*

corpus subset	# sentences (fraction)	# spoken words	WER[%], loss: sent.	words
all	740 (100%)	12137	3.97	3.88
a)	401 (54%)	6189	1.16	
b)	57 (8%)	990	3.64	
c)	229 (31%)	4023	6.56	
d)	53 (7%)	935	11.8	10.6

Nevertheless, it is interesting that this small improvement is obtained only from those 7% of the utterances, for which word error minimization gives a result different from sentence error minimization. For this fraction of the test utterances a relative improvement of about 7% in word error rate is obtained, cf. condition d) in Table 1. It is also interesting to notice the individual error rates calculated for the different conditions presented in Table 1. Particularly utterances which have very high posterior probability ($> 1/2$) also have a very low error rate.

The average sentence length here is nearly 18 words, and the baseline word error rate for this task is 3.97%. From the rough estimate presented at the end of Sec. 3.1, we would expect a word error rate of more than 4% to see significant differences in the decisions made by word and sentence error minimization. Therefore, the marginal improvement obtained here using word error minimization can be expected.

Finally, the average difference between the *Levenshtein* distance between the posterior maximizing word sequence and the *Bayes* risk minimizing word sequence, and the parameter ϵ derived in Eq.(12) is only 0.455, i.e. about one word in every second sentence. Therefore ϵ can be used to find a close bound to the difference between the *Bayes* risk for the posterior maximizing word sequence and the minimum *Bayes* risk and therefore allows for finding a good initial over-estimate of the *Bayes* risk via Ineq. (13). In addition, ϵ seems to be well suited in giving an efficiently calculable estimate of the potential change in word error rate when doing word error minimization instead of sentence error minimization.

Acknowledgements This work was partly funded by the European Union under the Integrated Project "TC-STAR - Technology and Corpora for Speech to Speech Translation" (IST-2002-FP6-506738, http://www.tc-star.org).

5. References

[1] V. Goel, W. Byrne: "Minimum Bayes Risk Automatic Speech Recognition," *Computer Speech and Language*, Vol. 14, No. 2, pp. 115–135, 2000.

[2] V. Goel, W. Byrne: "Minimum Bayes Risk Methods in Automatic Speech Recognition," in W. Chou, B.H. Juang (eds.): *Pattern Recognition in Speech and Language Processing*, pp. 51–80, CRC Press, Boca Raton, FL, 2003.

[3] W.Macherey, L. Haferkamp, R. Schlüter, H. Ney: "Investigations on Error Minimizing Training Criteria for Discriminative Training in Automatic Speech Recognition," submitted to *European Conference on Speech Communication and Technology (Interspeech)*, Lisbon, September 2005.

[4] L. Mangu, E. Brill, A. Stolcke: "Finding Consensus Among Words: Lattice-Based Word Error Minimization," *Proc. European Conference on Speech Communication and Technology (EUROSPEECH)*, pp. 495–498, Budapest, Hungary, Sept. 1999.

[5] A. Stolcke, Y. König, M. Weintraub: "Explicit Word Error Rate Minimization in N-Best List Rescoring," *Proc. European Conference on Speech Communication and Technology (EUROSPEECH)*, pp. 163–166, Rhodes, Greece, Sept. 1997.

[6] F. Wessel, R. Schlüter, H. Ney: "Explicit Word Error Minimization using Word Hypothesis Posterior Probabilities," Proc. *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 33-36, Salt Lake City, Utah, May 2001.